Applying Policers

Policers allow you to perform simple traffic policing on specific interfaces or Layer 2 virtual private networks (VPNs) without configuring a firewall filter. To apply policers, include the policer statement:

```
[Unresolved xref] {
    arp policer-template-name;
    input policer-template-name;
    output policer-template-name;
}
```

You can include these statements at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number* family *family*]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family *family*]

In the family statement, the protocol family can be ccc, inet, inet6, mpls, tcc, or vpls.

In the arp statement, list the name of one policer template to be evaluated when Address Resolution Protocol (ARP) packets are received on the interface. By default, an ARP policer is installed that is shared among all the Ethernet interfaces on which you have configured the family inet statement. If you want more stringent or lenient policing of ARP packets, you can configure an interface-specific policer and apply it to the interface. You configure an ARP policer just as you would configure any other policer, at the [edit firewall policer] hierarchy level. If you apply this policer to an interface, the default ARP packet policer is overridden. If you delete this policer, the default policer takes effect again.

In the input statement, list the name of one policer template to be evaluated when packets are received on the interface.

In the output statement, list the name of one policer template to be evaluated when packets are transmitted on the interface.

NOTE: To use policing on a CCC or TCC interface, you must configure the CCC or TCC protocol family.

You can configure a different policer on each protocol family on an interface, with one input policer and one output policer for each family. When you apply policers, you can configure the family ccc, inet, inet6, mpls, tcc, or vpls only, and one ARP policer for the family inet protocol only. Each time a policer is referenced, a separate copy of the policer is installed on the packet forwarding components for that interface.

If you apply both policers and firewall filters to an interface, input policers are evaluated before input firewall filters, and output policers are evaluated after output firewall filters.

If you apply the policer to the interface lo0, it is applied to packets received or transmitted by the Routing Engine.

On M-series platforms (except the M320 and M120 routers), if you apply a firewall filter or policer to multiple interfaces, the filter or policer acts on the sum of traffic entering or exiting those interfaces. On T-series, M120, and M320 platforms, the filter or policer acts on the sum of traffic, if the interfaces are on the same FPC.

For more information about policers, see the *JUNOS Policy Framework Configuration Guide*.

Applying Aggregate Policers

By default, if you apply a policer to multiple protocol families on the same logical interface, the policer restricts traffic for each protocol family individually. For example, a policer with a 50 Mbps bandwidth limit applied to both IPv4 and IPv6 traffic would allow the interface to accept 50 Mbps of IPv4 traffic and 50 Mbps of IPv6 traffic. If you apply an aggregate policer, the policer would allow the interface to receive only 50 Mbps of IPv4 and IPv6 traffic combined.

To configure an aggregate policer, include the logical-interface-policer statement at the [edit firewall policer *policer-template-name*] hierarchy level:

```
[edit firewall policer policer-template-name]
logical-interface-policer;
```

For the policer to be treated as an aggregate, you must apply it to multiple protocol families on a single logical interface by including the policer statement:

```
[Unresolved xref] {
    arp policer-template-name;
    input policer-template-name;
    output policer-template-name;
}
```

You can include these statements at the following hierarchy levels:

- [edit interfaces *interface-name* unit *logical-unit-number* family *family*]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family *family*]

In the family statement, the protocol family can be ccc, inet, inet6, mpls, tcc, or vpls.

The protocol families on which you do not apply the policer are not affected by the policer. For example, if you configure a single logical interface to accept MPLS, IPv4, and IPv6 traffic and you apply the logical interface policer policer1 to only the IPv4 and IPv6 protocol families, MPLS traffic is not subject to the constraints of policer1.

If you apply policer1 to a different logical interface, there are two instances of the policer. This means the JUNOS software polices traffic on separate logical interfaces separately, not as an aggregate, even if the same logical-interface policer is applied to multiple logical interfaces on the same physical interface port.

NOTE: Logical interface policers are not supported for filter policers. In other words, you cannot include the logical-interface-policer statement at the [edit firewall filter *name* term *name* then policer] hierarchy level.

Example: Applying Aggregate Policers

Configure two logical interface policers: aggregate_police1 and aggregate_police2. Apply aggregate_police1 to IPv4 and IPv6 traffic received on logical interface fe-0/0/0.0. Apply aggregate_police2 to CCC and MPLS traffic received on logical interface fe-0/0/0.0. This configuration causes the software to create only one instance of aggregate_police1 and one instance of aggregate_police2.

Apply aggregate_police1 to IPv4 and IPv6 traffic received on another logical interface fe-0/0/0.1. This configuration causes the software to create a new instance of aggregate_police1, one that applies to unit 0 and another that applies to unit 1.

```
[edit firewall]
policer aggregate_police1 {
    logical-interface-policer;
    if-exceeding {
        bandwidth-limit 100m;
        burst-size-limit 500k;
    }
    then {
        discard;
    }
}
policer aggregate_police2 {
    logical-interface-policer;
    if-exceeding {
        bandwidth-limit 10m;
        burst-size-limit 200k;
    }
    then {
        discard;
    }
}
[edit interfaces fe-0/0/0]
unit 0 {
    family inet {
        policer {
            input aggregate_police1;
        }
    }
    family inet6 {
        policer {
            input aggregate_police1;
        }
    }
    family ccc {
        policer {
            input aggregate_police2;
```

```
        }
    }
    family mpls {
        policer {
            input aggregate_police2;
        }
    }
}
unit 1 {
    family inet {
        policer {
            input aggregate_police1;
        }
    }
    family inet6 {
        policer {
            input aggregate_police1;
        }
    }
}
```

Applying Hierarchical Policers on Enhanced Intelligent Queuing PICs

M40e, M120, and M320 edge routers and T-series core routers with Enhanced Intelligent Queuing (IQE) PICs support hierarchical policers in the ingress direction and allow you to apply a hierarchical policer for the premium and aggregate (premium plus normal) traffic levels to an interface. Hierarchical policers provide cross-functionality between the configured physical interface and the packet forwarding engine (PFE).

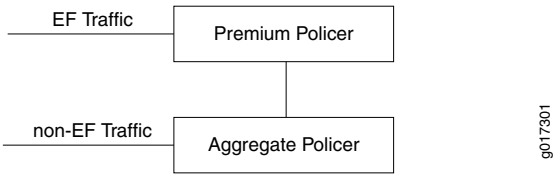
Before you begin, there are some general restrictions that apply to hierarchical policers:

- Only one type of policer can be configured for a logical or physical interface. For example, a hierarchical policer and a regular policer in the same direction for the same logical interface is not allowed.
- The chaining of the policers - that is, applying policers to both a port and the logical interfaces of that port - is not allowed.
- There is a limit of 64 policers per interface in case there is no BA classification, providing a single policer per DLCI.
- Only one kind of policer can be applied on a physical or logical interface.
- The policer should be independent of BA classification. Without BA classification, all traffic on an interface will be treated either as EF or non-EF, based on the configuration. With BA classification, an interface can support up to 64 policers. Again, the interface here may be a physical interface or logical interface (i.e. DLCI).
- With BA classification, the miscellaneous traffic (i.e. the traffic *not* matching with any of the BA classification DSCP/EXP bits) will be policed as non-EF traffic. No separate policers will be installed for this traffic.

Hierarchical Policer Overview

Hierarchical policing uses two token buckets, one for aggregate (non-EF) traffic and one for premium (EF) traffic. Which traffic is EF and which is non-EF is determined by the class-of-service configuration. Logically, hierarchical policing is achieved by chaining two policers.

Figure 1: Hierarchical Policer



In the example in Figure 1, EF traffic is policed by "Premium Policer" and non EF traffic is policed by "Aggregate Policer". What that means is, for EF traffic the out-of-spec action will be the one that is configured for "Premium Policer", but the in-spec EF traffic will still consume the tokens from the "Aggregate Policer".

But EF traffic will never be submitted to the out-of-spec action of the "Aggregate Policer". Also, if the out-of-spec action of the "Premium Policer" is not set to "Discard", those out-of-spec packets will not consume the tokens from the "Aggregate Policer". "Aggregate Policer" only polices the non-EF traffic. As you can see, the "Aggregate Policer" token bucket can go negative, if all the tokens are consumed by the non-EF traffic and then you get bursts of EF traffic. But that will be for a very short time, and over a period of time it will average out. Let's take an example:

- *Premium Policer:* Bandwidth 2 Mbps, OOS Action: Discard
- *Aggregate Policer:* Bandwidth 10 Mbps, OOS Action: Discard

In the above case, EF traffic is guaranteed 2 Mbps and the non-EF traffic will get from 8 Mbps to 10 Mbps, depending on the input rate of the EF traffic.

Hierarchical Policing Characteristics

Hierarchical Token Buckets

- Ingress traffic is first classified into EF and non-EF traffic prior to applying a policer:
  - Classification is performed by Q-tree look-up
- Channel Number selects a Shared Token Bucket Policer:
  - Dual token bucker policer is divided into (2) single bucket policers:
    - Policer1 - EF traffic
    - Policer2 - non-EF traffic
- Shared-token bucket is used to police the traffic as follows:

- Policer1 is set to EF rate (e.g., 2 Mbps)
- Policer2 is set to aggregate interface policed rate (e.g., 10 Mbps).
- EF traffic gets applied to Policer1
  - If traffic is in-spec it is allowed to pass and decrement from both Policer1 and Policer2
  - If traffic is out-of-spec it can be discarded or marked with a new FC or loss priority. Policer2 will not do anything with out-of-spec EF traffic.
- Non-EF traffic gets applied only to Policer2
  - If traffic is in-spec it is allowed to pass through and decremented Policer2
  - If traffic is out-of-spec it is discarded or marked with a new FC or set with a new drop priority

- Rate-limit the port speed to a desired rate at Layer 2
- Rate-limit the EF traffic
- Rate-limit the non-EF traffic
- Policing drops counted per color

Configuring Hierarchical Policers

To configure a hierarchical policer, apply the policing-priority statement to the proper forwarding class and configure a hierarchical policer for the aggregate and premium level. For more information on Class of Service configuration, see *Class of Service*.

NOTE: Hierarchical policers can only be configured on SONET physical interfaces hosted on an IQE PIC. Only aggregate and premium levels are supported.

CoS configuration of forwarding-class stanza for hierarchical policers

```
[edit class-of-service forwarding-classes]
class fc1 queue-num 0 priority high policing-priority premium
class fc2 queue-num 1 priority low policing-priority normal
class fc3 queue-num 2 priority low policing-priority normal
class fc4 queue-num 3 priority low policing-priority normal
```

For detailed information on Class of Service configuration and statements, see Class of Service.

Firewall configuration for hierarchical policers

```
[edit firewall hierarchical-policer foo]
aggregate {
   if-exceeding {
      bandwidth-limit 70m;
      burst-size-limit 1500;
   }
   then {
   discard;
}
premium {
```

```
    if-exceeding {
        bandwidth-limit 50m;
        burst-size-limit 1500;
    }
    then {
        discard;
    }
}
```

You can apply the hierarchical policer as follows:

```
[edit interfaces so-0/1/0 unit 0 layer-2-policer]
input-hierarchical-policer foo
```

You also have the option to apply the policer at the physical port level as follows:

```
[edit interfaces so-0/1/0 layer-2-policer]
input-hierarchical-policer foo
```

Configuring a Single-Rate Two-Color Policer

You can configure a single-rate two-color policer as follows:

Firewall configuration for two-color policer

```
[edit firewall policer foo]
    if-exceeding {
        bandwidth-limit 50m;
        burst-size-limit 1500;
    }
then {
    discard;
}
```

You can apply the policer as follows:

```
[edit interfaces so-0/1/0 unit 0 layer-2-policer]
input-policer foo
```

You also have the option to apply the policer at the physical port level as follows:

```
[edit interfaces so-0/1/0 layer-2-policer]
input-policer foo
```

Configuring a Single-Rate Tri-Color Policer

This section describes single-rate color-blind and color-aware policers.

Configuring a Single-Rate Color-Blind Policer

You can configure a single-rate color-blind policer as follows:

Firewall configuration for a single-rate color-blind policer

```
[edit firewall three-color-policer foo]
single-rate {
    color-blind;
    committed-information-rate 50m;
    committed-burst-size 1500;
```

```
    excess-burst-size 1500;
}
```

You can apply the single-rate color-blind policer as follows:

```
[edit interfaces so-0/1/0 unit 0 layer-2-policer]
input-three-color foo
```

You also have the option to apply the policer at the physical port level as follows:

```
[edit interfaces so-0/1/0 layer-2-policer]
input-three-color foo
```

Configuring a Single-Rate Color-Aware Policer

You can configure a single-rate color-aware policer as follows:

Firewall configuration for a single-rate color-aware policer

```
[edit firewall three-color-policer bar]
single-rate {
    color-aware;
    committed-information-rate 50m;
    committed-burst-size 1500;
    excess-burst-size 1500;
}
```

You can apply the single-rate color-aware policer as follows:

```
[edit interfaces so-0/1/0 unit 0 layer-2-policer]
input-three-color foo
```

You also have the option to apply the policer at the physical port level as follows:

```
[edit interfaces so-0/1/0 layer-2-policer]
input-three-color bar
```

Configuring a Two-Rate Tri-Color Marker Policer

Ingress policing is implemented using a two-rate tri-color marker (trTCM). This is done with a dual token bucket (DTB) that maintains two rates, committed, and a peak. Egress static policing also uses a token bucket.

The token buckets perform the following Ingress Policing functions:

- (1K) trTCM - Dual Token Bucket (Red, Yellow, and Green marking)
- Policing s based on L2 packet size
  - After +/- Byte Adjust Offset
- Marking is Color Aware and Color Blind
  - Color Aware needs to have the Color set by Q-tree look-up based on
    - ToS
    - EXP
- Programmable Marking Actions

- Color (Red, Yellow, Green)
- Drop based on color and congestion profile

- Policer is selected based on the arriving Channel Number
  - Channel number LUT produces Policer index and Queue index
  - Multiple channels can share the same policer (LUT produces same policer index)

- Support ingress policing and trTCM at the following levels:
  - Queue
  - Logical Interface (aka ifl/DLCI)
  - Physical Interface (aka ifd)
  - Physical Port (aka controller ifd)
  - Any combinations of logical interface, physical interface and port

- Support percentage of interface speed and bits per second

Rate limits may be applied to selected queues on ingress and on predefined queues at egress. The token bucket operates in color aware and color blind modes (specified by RFC 2698).

Color Blind

```
[edit firewall three-color-policer foo]
two-rate {
   color-blind;
   committed-information-rate 50m;
   committed-burst-size 1500;
   peak-information-rate 100m;
   peak-burst-size 3k;
}
```

You can apply the three-color two-rate color-blind policer as follows:

```
[edit interfaces so-0/1/0 unit 0 layer-2-policer]
input-three-color foo
```

You also have the option to apply the policer at the physical port level as follows:

```
[edit interfaces so-0/1/0 layer-2-policer]
input-three-color foo
```

Color Aware

```
[edit firewall three-color-policer bar]
two-rate {
   color-aware;
   committed-information-rate 50m;
   committed-burst-size 1500;
   peak-information-rate 100m;
   peak-burst-size 3k;
}
```

You can apply the three-color two-rate color-aware policer as follows:

```
[edit interfaces so-0/1/0 unit 0 layer-2-policer]
input-three-color bar
```

You also have the option to apply the policer at the physical port level as follows:

```
[edit interfaces so-0/1/0 layer-2-policer]
input-three-color bar
```